



Java and Wireless

An Embedded Challenge

Ralf Facht, Wind River Central Europe
June 2002



Agenda:

□ Wind River

- Short overview
- Java Expertise

□ Java and Wireless

- Wireless Protocols
- Embedded Applications

□ Summary

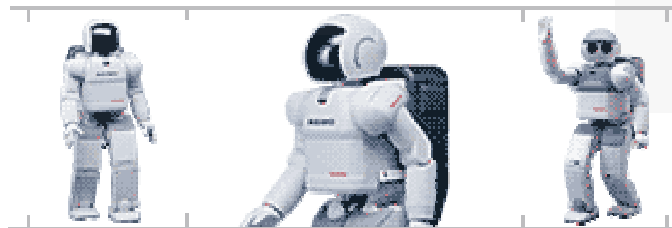


Company Overview



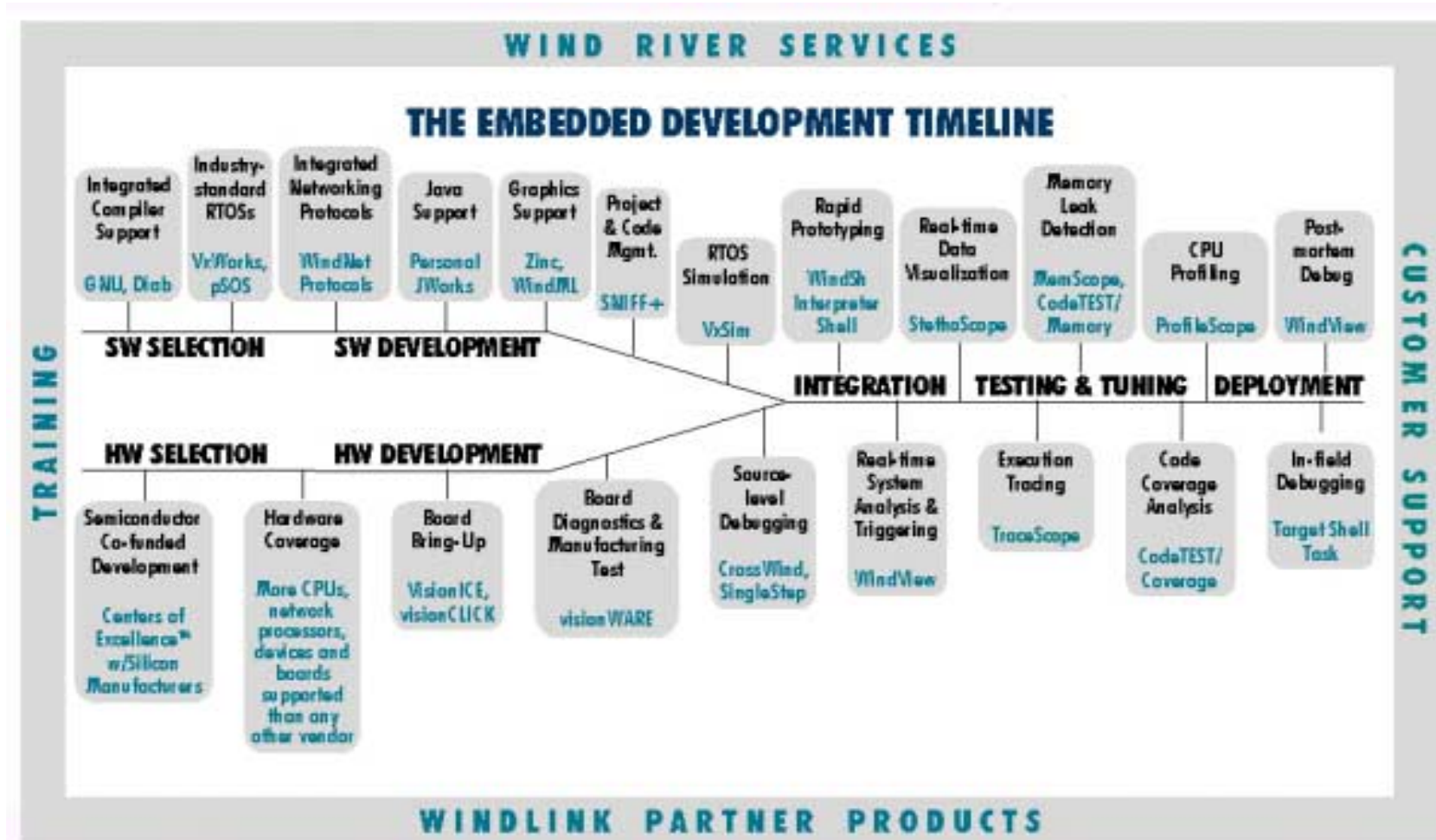
WindRiver - Experts for Embedded Systems

- ❑ **1.450 Employees**
 - 450 R&D Engineers
 - 450 Field Personnel
 - 250 Support Engineers
 - 500 Partners
- ❑ **50.000 Developers**
- ❑ **10.000 Active Projects**
- ❑ **100.000.000 Units Deployed**



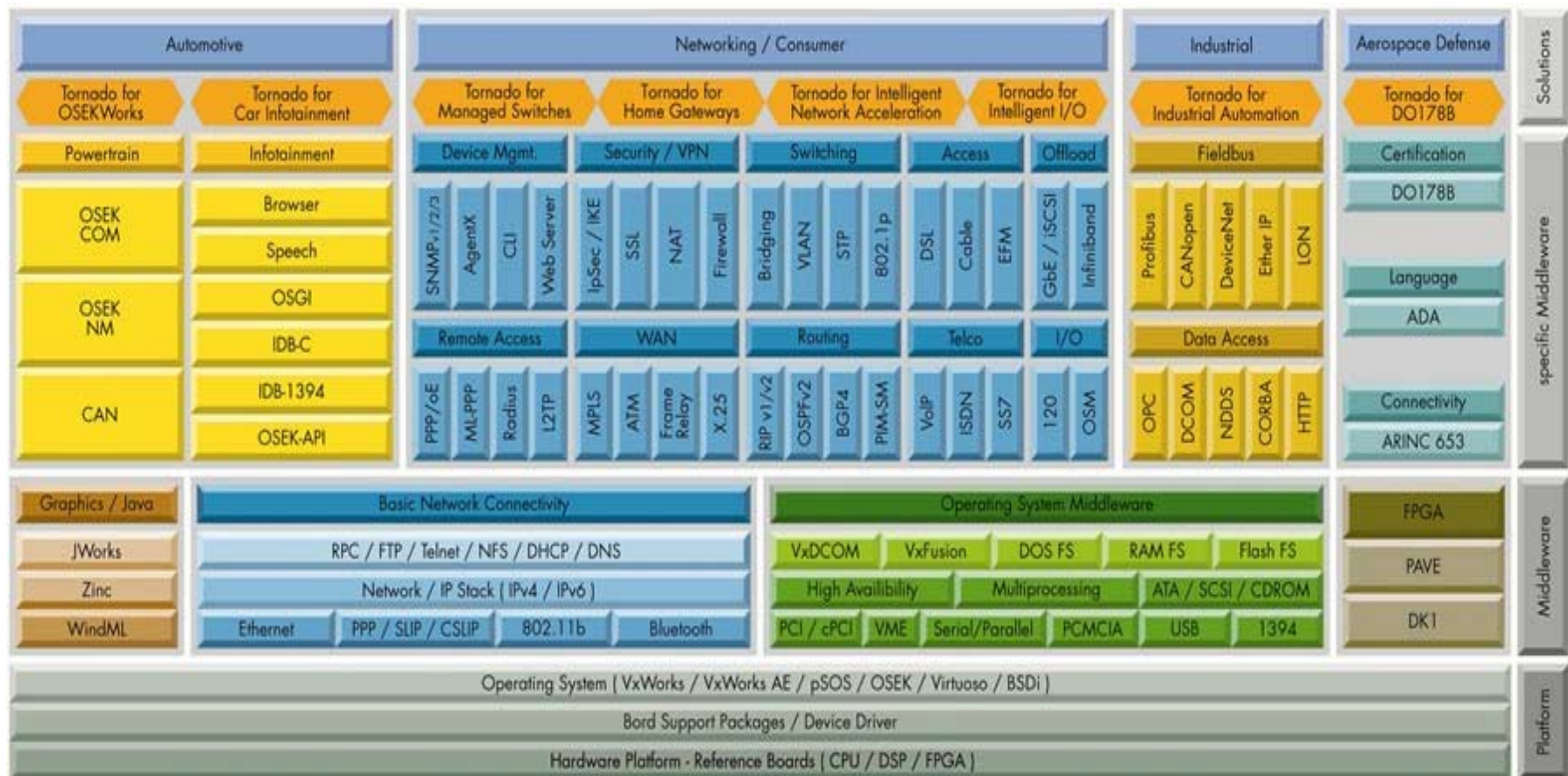


Life Cycle Solution





Wind River Product Portfolio





WindRiver and Java



A Rich History

□ Wind River & Java

- Started working on Java in 1995
- First to achieve fully compatible JDK 1.1 port to an embedded RTOS
- First to offer commercial certified PersonalJava product



□ Personal JWorks

- Sun's PersonalJava port to VxWorks
- Performance & graphics enhancements
- Foundation for application platform
- 3 releases – about 250 projects and more than 500K units shipped



□ Wind River & Insignia

- Strategic partnership entered in December 2000
- Integration of Jeode technology to next generation Java products





History of Successes

Public references

- GTECH Lottery terminal
- Invensys Flow meter
- Tridium Industrial controller
- Tadpole Rugged pen-computer
- Tektronix Oscilloscope
- GoReader E-book
- Pingtel Web phone
- Mitsubishi HDTV



*Tadpole J-Slate
Industrial Tablet*



*GoReader
Internet eBook*



*Samsung AnyWeb
Internet Screen Phone*



*Tektronix TDS7000
Digital Phosphor
Oscilloscopes*



*GTECH Altura
Java Lottery Terminal*



*eRemote
Universal Remote Control*

And more...

- Acterna, Agilent, Alcatel, Bull, Compuware, Daewoo, Intel, Kenwood, Lucent, Motorola, Nortel, Philips, Samsung, Siemens, Sony, etc.



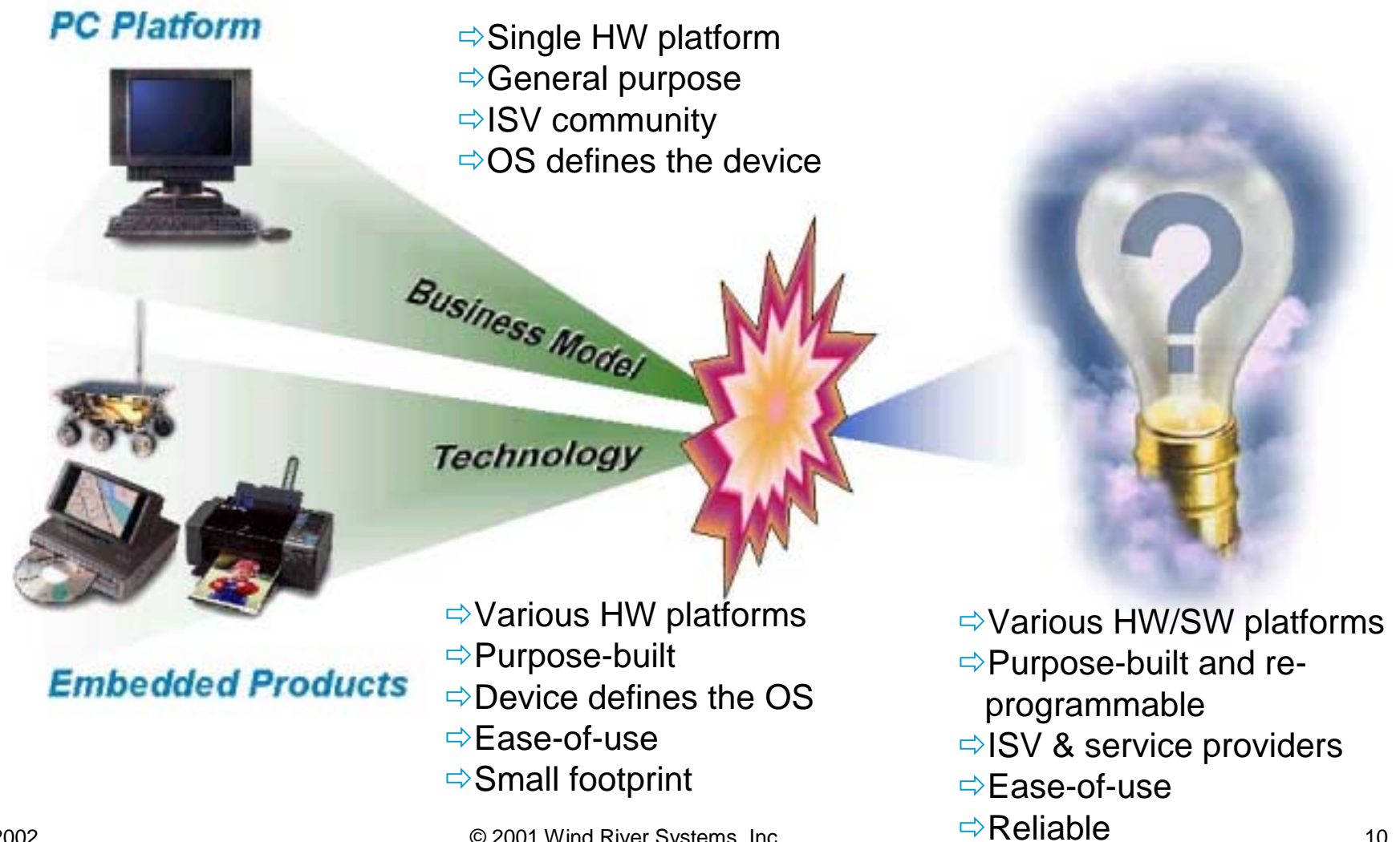
*PingTel Xpressa
VOIP Screen Phone*



*Intel Web Tablet
Internet Appliance*



Convergence of PC & Embedded





The Role of Java in Embedded

□ Enabling the new generation of embedded platforms

- Connected, Internet-ready
- Reliable and secure
- Interactive
- Re-programmable

□ Universal application & service delivery platform

- Dynamic extensibility → re-programmability
- Standardized API & platform independence → ISV community

□ Development paradigm

- Simple and efficient OO programming language
- Large base of content developers for investment protection

□ Custom branding & look-and-feel





A Wireless Application

□ Information Services ...

- Retrieve shopping list via a smart phone while on the go!

□ Security - Controlling Access

- Controlling access
 - Send one-time-use entry code to client's mobile
 - Client uses mobile to unlock the door

□ Home Monitoring & Control

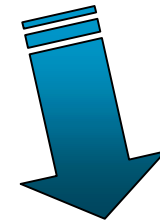
- Camera/phone server controlled by gateway
 - Audio & low resolution images beamed to mobile phone
 - Can automatically send wireless messages on event detection
- Java application on phone to control home
 - Turn on lights
 - Put the cat out



*CMI Digital IceBox
Kitchen Appliance*



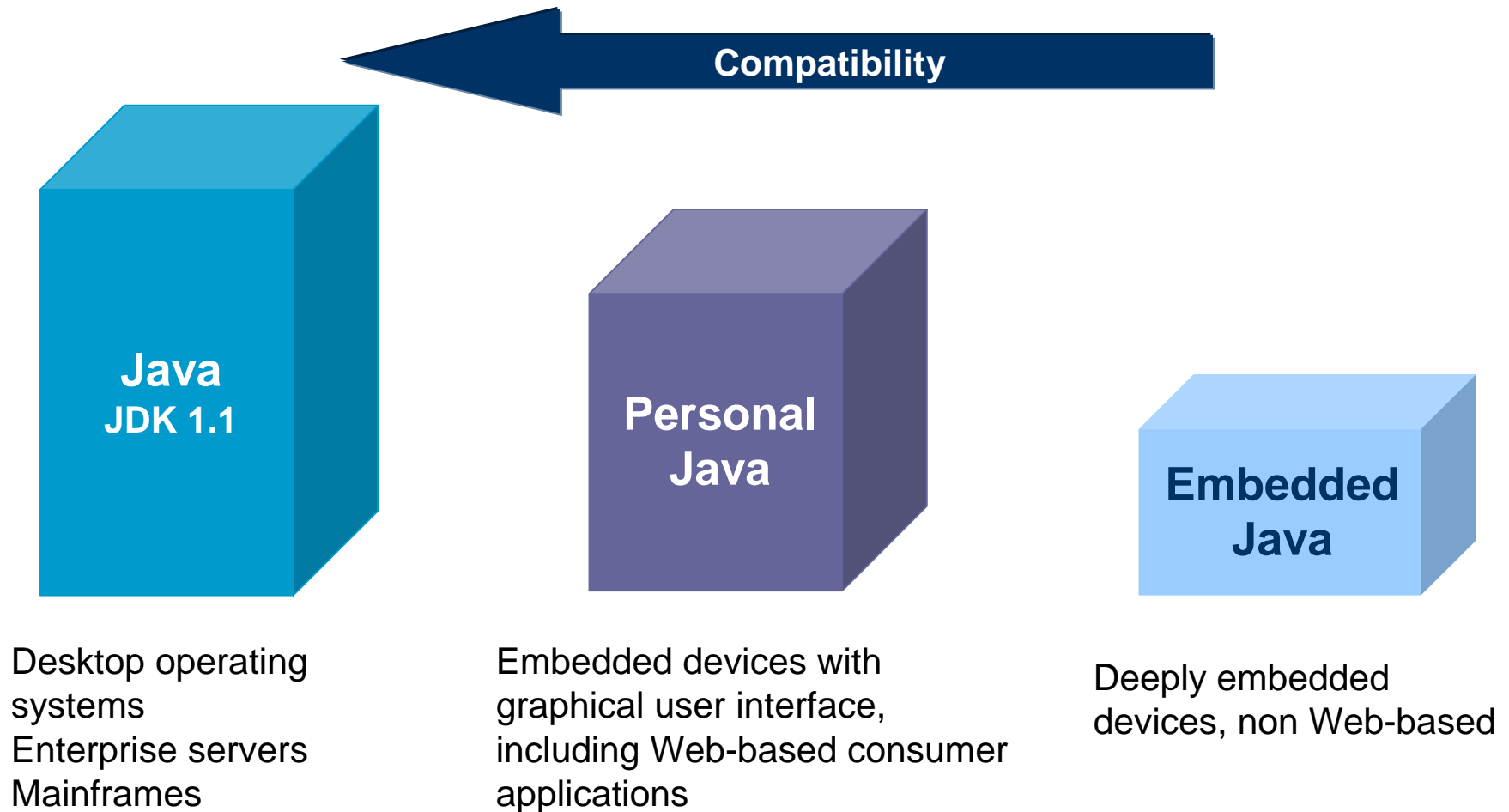
*2Wire
HomePort 1000
Residential
Gateway*



*Mitsubishi Internet Paldio
PHS HTML Smart Phone*

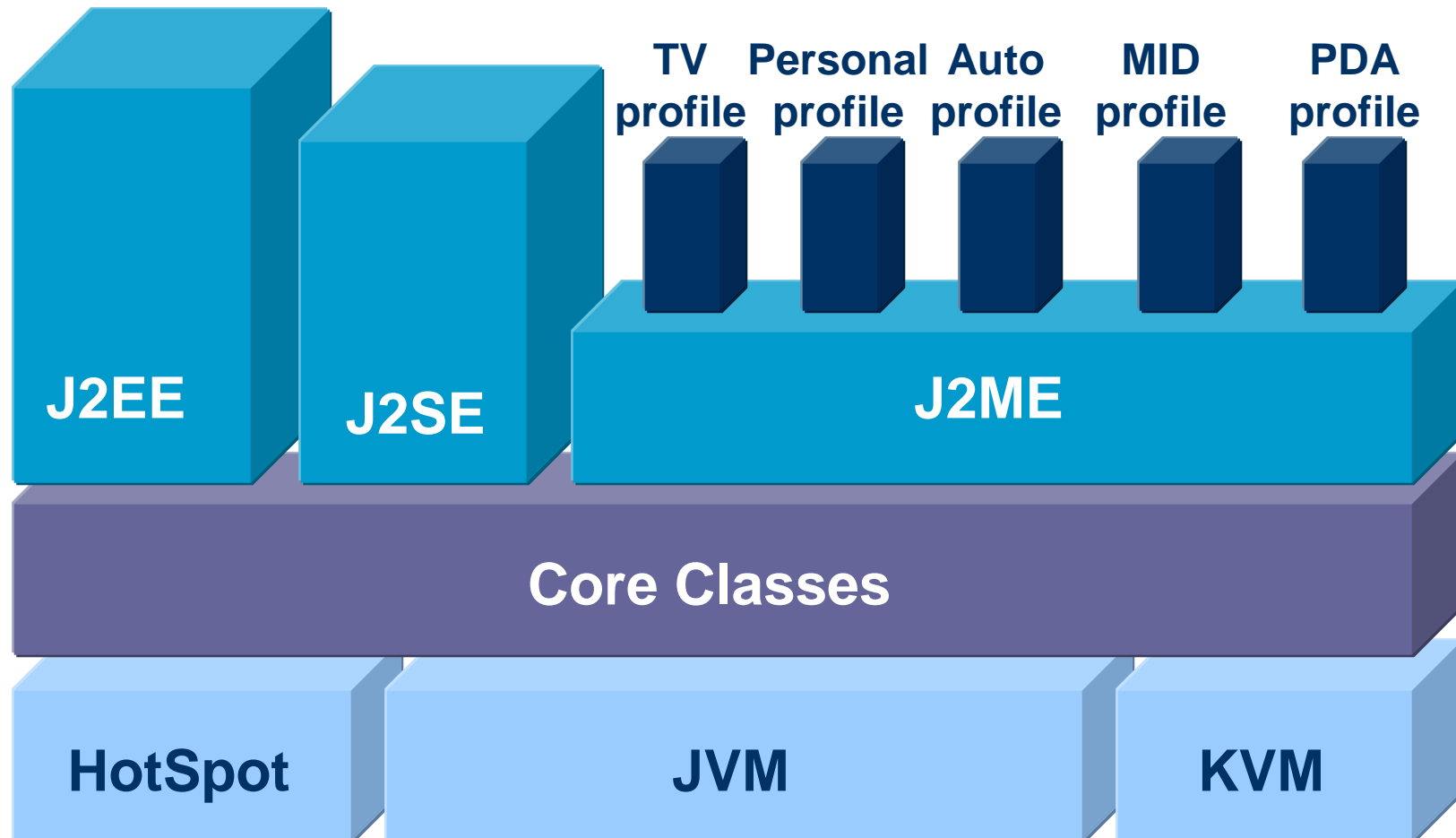


Java Application Environments



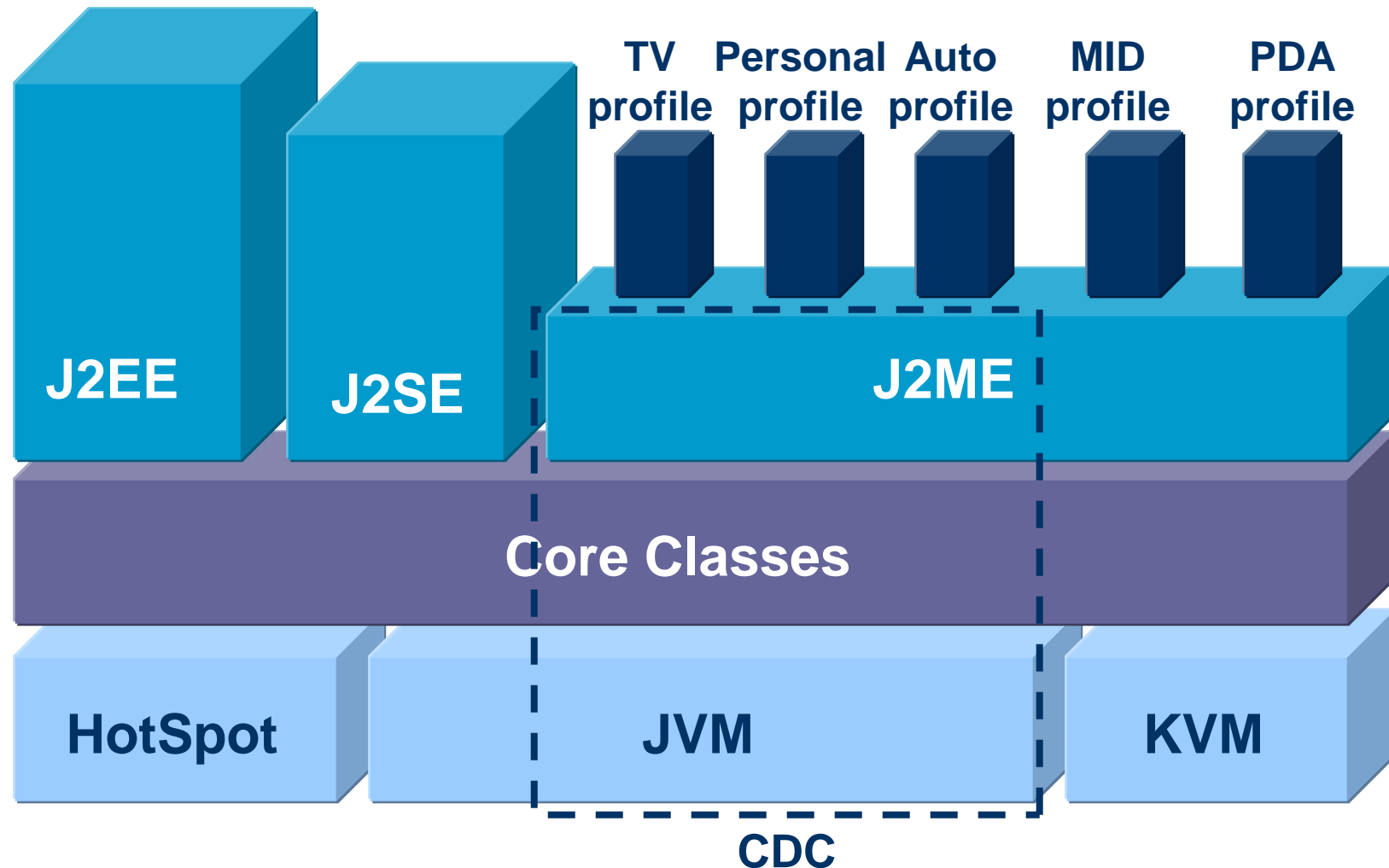


Java 2 Configurations & Profiles





Connected Device Configuration





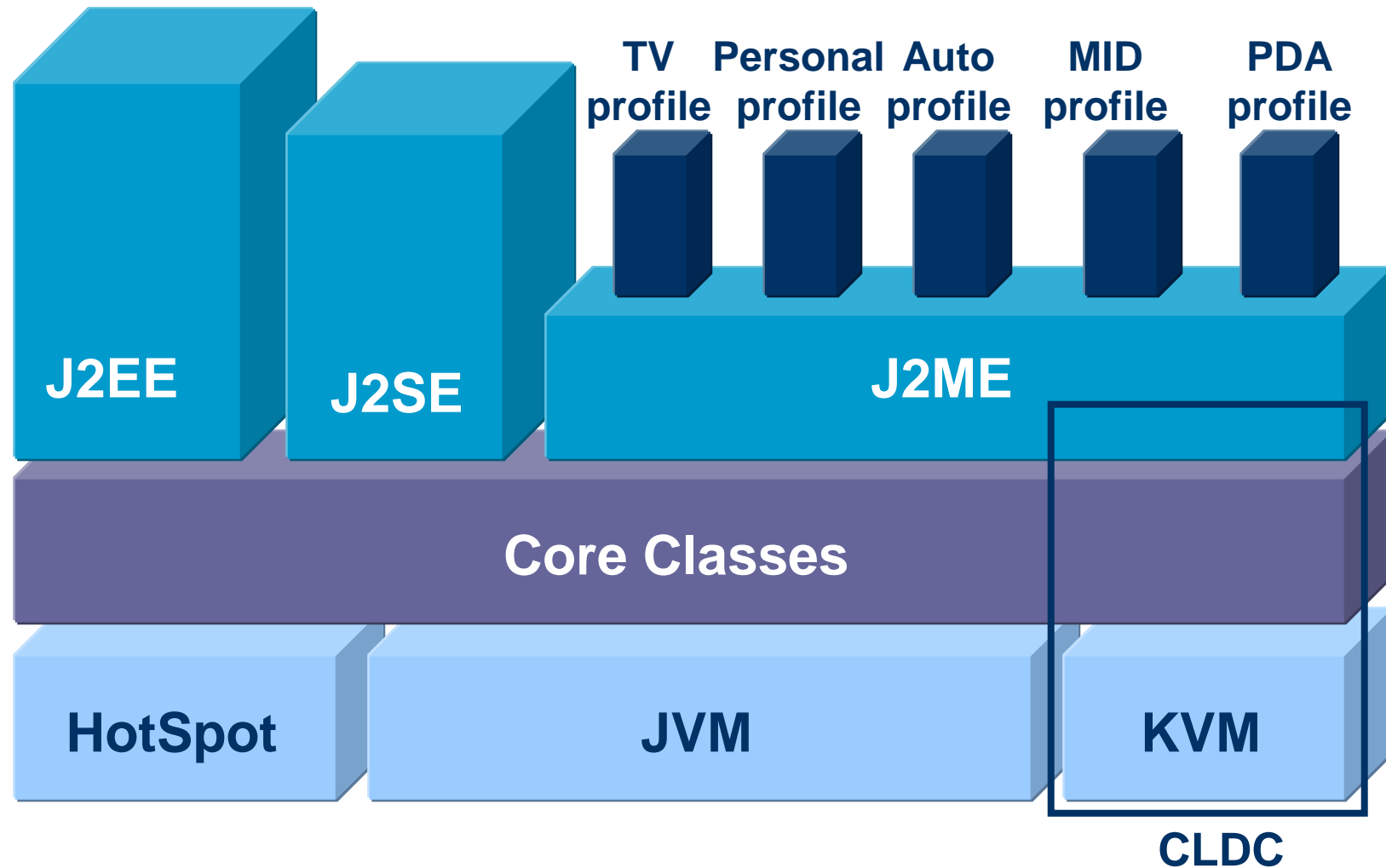
J2ME CDC

□ Connected Device Configuration (CDC)

- 32-bit full featured JVM
- > 2 MB of memory available for Java
- Connectivity to some kind of network
- Relies on underlying RTOS and C runtime environment
- API = subset of J2SE + specific classes
- User interfaces (still under specification)
 - Varying degrees of sophistication or even none
 - Personal Profile = Foundation Profile + PersonalJava's graphics updated to SDK 1.3
 - PersonalBasis Profile = Foundation Profile + small subset of AWT (suitable for TV)
 - Extensions not specified in profiles: JavaTV, HAVi, Java 2D/3D, etc.
- <http://www.sun.com/software/communitysource/j2me/cdc/>



Connected Limited Device Configuration





J2ME CLDC

□ Connected Limited Device Configuration (CLDC)

- 16- or 32-bit KVM
- 160 to 512 KB of memory available for Java
- Does not necessarily require a “persistent” network connection
 - Typically wireless, intermittent connection
 - Limited (9600- bps) bandwidth
- Limited power, typically battery powered operation
- Limitations on Java language and JVM features
- Run pre-verified classes
- APIs = subset of J2SE + specific classes
- User interfaces
 - Mobile information device (MID) profile: LCDUI
 - Personal data assistant (PDA) profile: subset of AWT
 - Palm’s kAWT
- <http://www.sun.com/software/communitysource/j2me/cldc/>

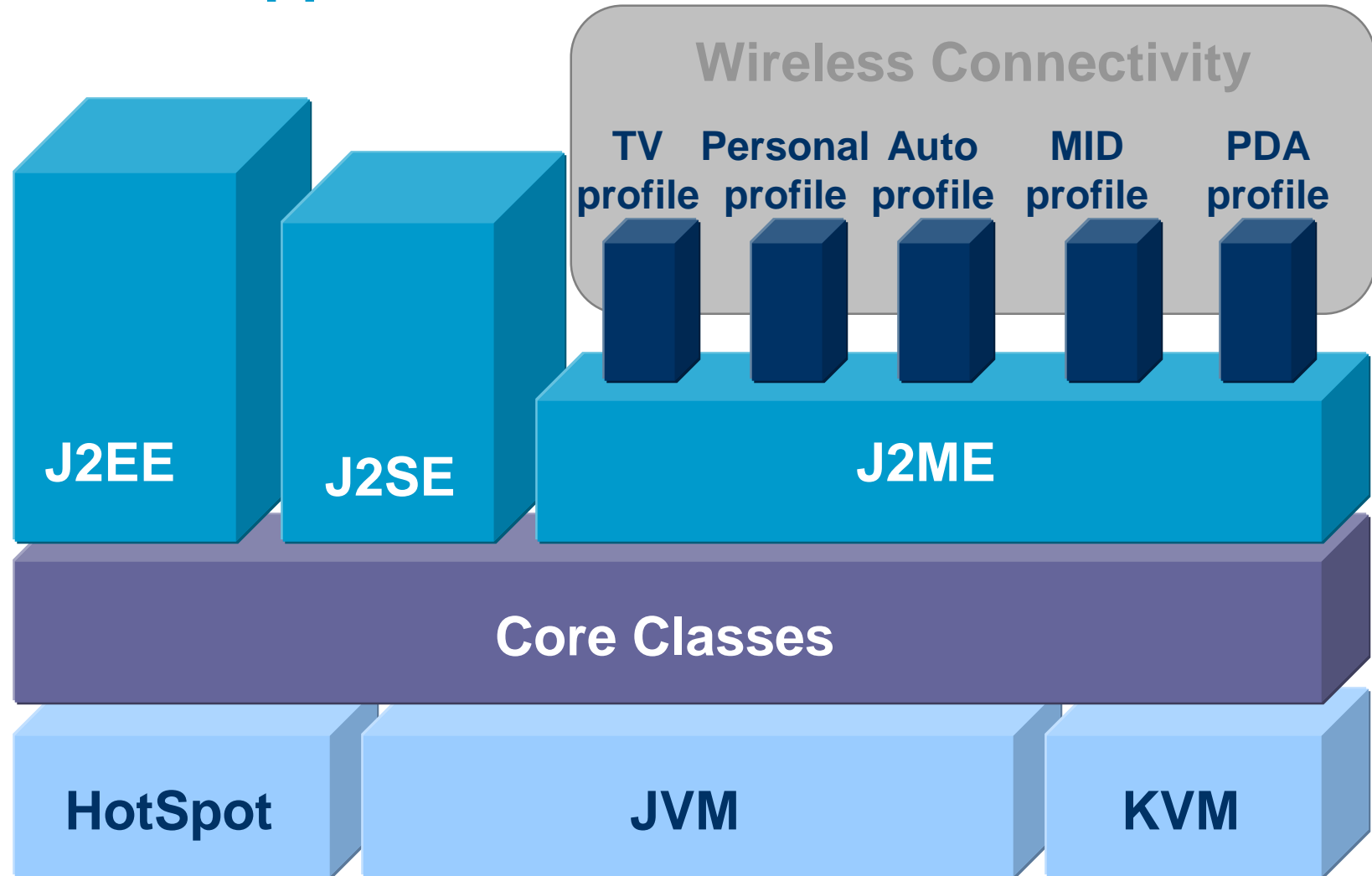


KVM Limitations

- ❑ No FP support
- ❑ No JNI
 - No target-based debug agent
- ❑ No user-defined class loaders
 - Can't run applet-style apps but only complete applications
- ❑ No reflection
 - No RMI
- ❑ No daemon thread
- ❑ No thread groups
- ❑ No finalization
- ❑ No weak references



Wireless Applications





Bluetooth

❑ Java APIs for Bluetooth Wireless Technology

- Java Specification Request 82 (JSR-82),
- Specification Lead Motorola

❑ Java Requirements

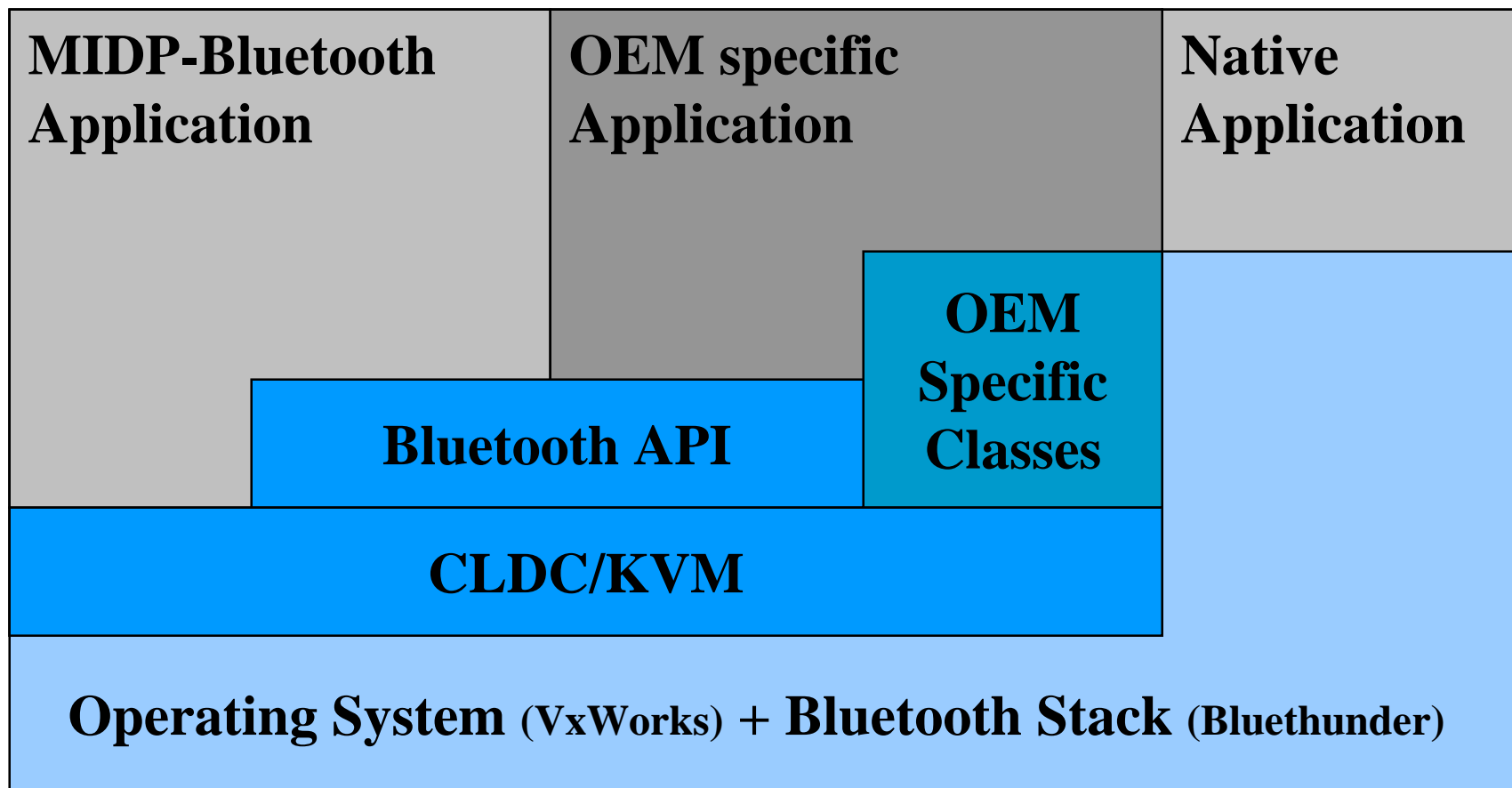
- Compliant implementation of the J2ME Connected Limited Device Configuration or a superset of CLDC APIs, such as the J2ME Connected Device Configuration (CDC)

❑ Bluetooth Requirements

- The following layers are supported as defined in the Bluetooth specification version 1.1, and the implementation of this API has access to them.
 - Service Discovery Protocol (SDP)
 - RFCOMM (type 1 device support)
 - Logical Link Control and Adaptation Protocol (L2CAP)



Bluetooth Architecture Diagram





Wireless Message Service

❑ Wireless Messaging API

- Java Specification Request 120 (JSR-120),
- Specification Lead by Siemens

❑ Addressed Technologies

- Short Message Service (SMS)
- Unstructured Supplementary Service Data (USSD)
- Cell Broadcast Service (CBS)

❑ Planned API for

- sending and receiving text (API)
- usage as data bearer (API)
- application push, connection to the JAM
- application trigger

❑ Schedule for the development of this specification

- Q3 2002

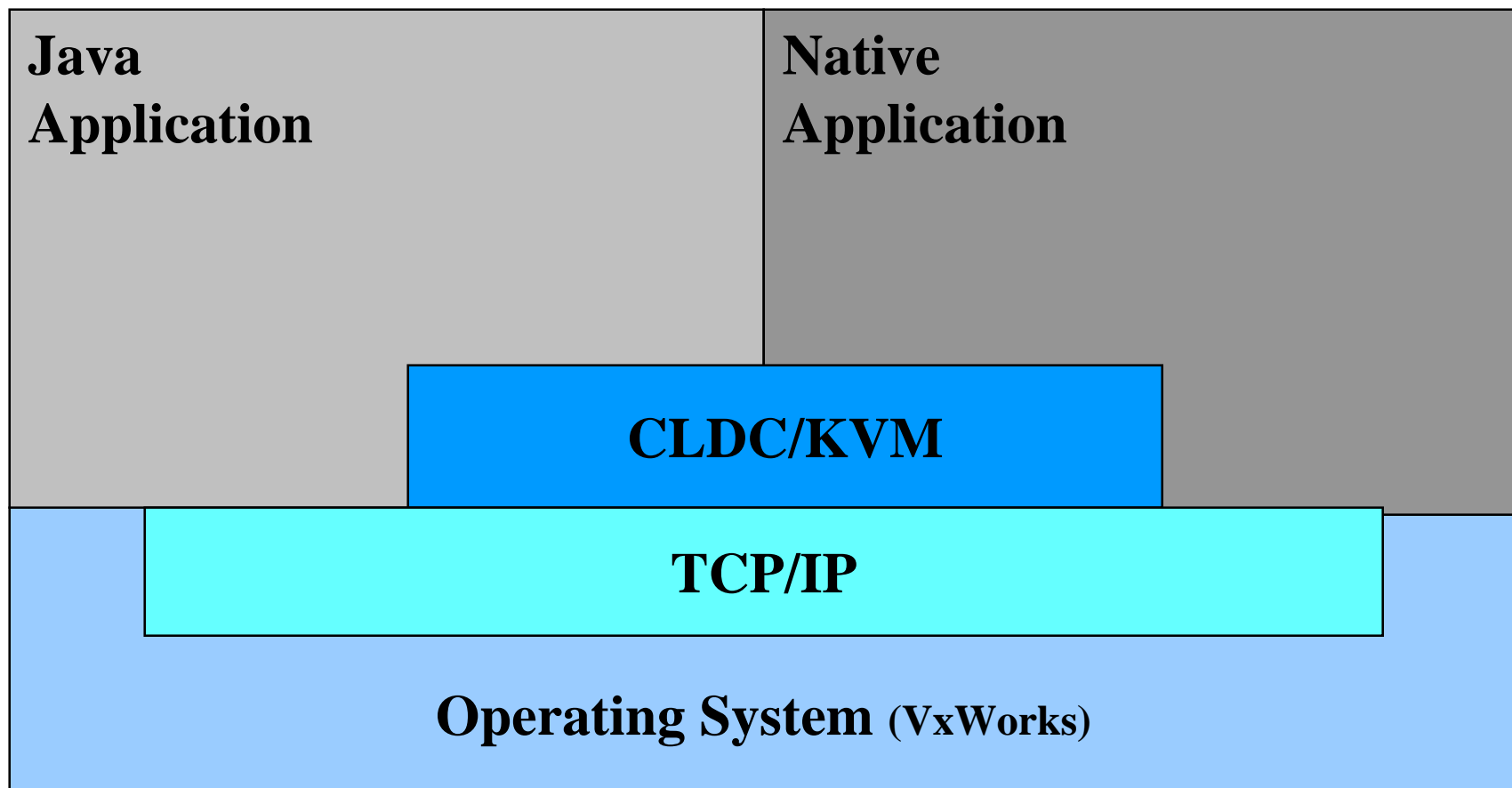


Wireless Lan - WLAN

- ❑ **Wireless Ethernet (IEEE 803.11)**
 - Defacto Standard for WLANs
 - <http://www.wirelessethernet.com/>
- ❑ **No special WLAN Java API**
 - Standard TCP/IP
- ❑ **Java Requirements**
 - ./.
- ❑ **WLAN Requirements**
 - ./.



(W)LAN Architecture Diagram





...but

Is the Protocol the „real“ Challenge in Wireless Java Applications ?



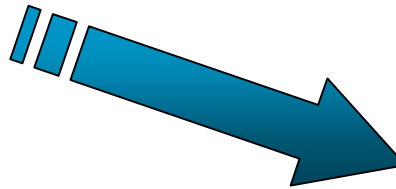
Another Wireless Scenario

- ❑ Online multi-player Gaming
- ❑ Digital postcards
- ❑ Mobile commerce
- ❑ Location-based services
- ❑ How Does Java Help?
 - Separating function from content
 - Enabling a client-server paradigm
 - High-level application coding
 - End-to-end security
- ❑ Simplification: Wireless = Embedded

*Samsung SCH-i201
CDMA HTML Smart Phone*



*Casio QVC3000
Digital Camera*



*Polaroid Ceiva
Internet Picture Frame*



*3Com BT-1000
Bluetooth Access Point*



Java Market Data

- ❑ **Programmers using Java: estimated 1 million**
 - Number of Java programmers surpassed C++ programmers in 2000
 - Will triple by 2002 (IDC)
- ❑ **44% of worldwide software developers are using Java (IDC)**
- ❑ **Java is taught at 78% of universities**
- ❑ **JVM penetration > 90% of desktop and servers by 2002 (Gartner)**

- ❑ **Present**
 - 10% of embedded developers use Java (EETimes)
- ❑ **Future**
 - 24.3M Java-powered devices shipping in 2003 (VDC)
 - 25% of embedded projects considering Java (CMP)



Embedded Java Challenges

- ❑ Non-deterministic Java application environment ✓
- ❑ Integration to embedded RTOS ✓
- ❑ Memory footprint ✓
- ❑ Graphic stack functionality and performance ✓
- ❑ Dynamic compilation technology ✓
- ❑ Garbage collection technology ✓
- ❑ **Danger = wrong expectations**
 - Adequate system resources required depending on Java application

	No Graphics	Limited Graphics	Full Web Browsing
Processors	> 100 MIPS	150 - 250 MIPS	250+ MIPS
RAM	> 4 MB	> 8 MB	> 16 MB
ROM	> 2 MB	> 4 MB	> 8MB



Java in Embedded – The Pitfall

❑ **Java concepts cannot be applied from desktop/enterprise to embedded without taking into account specifics of embedded**

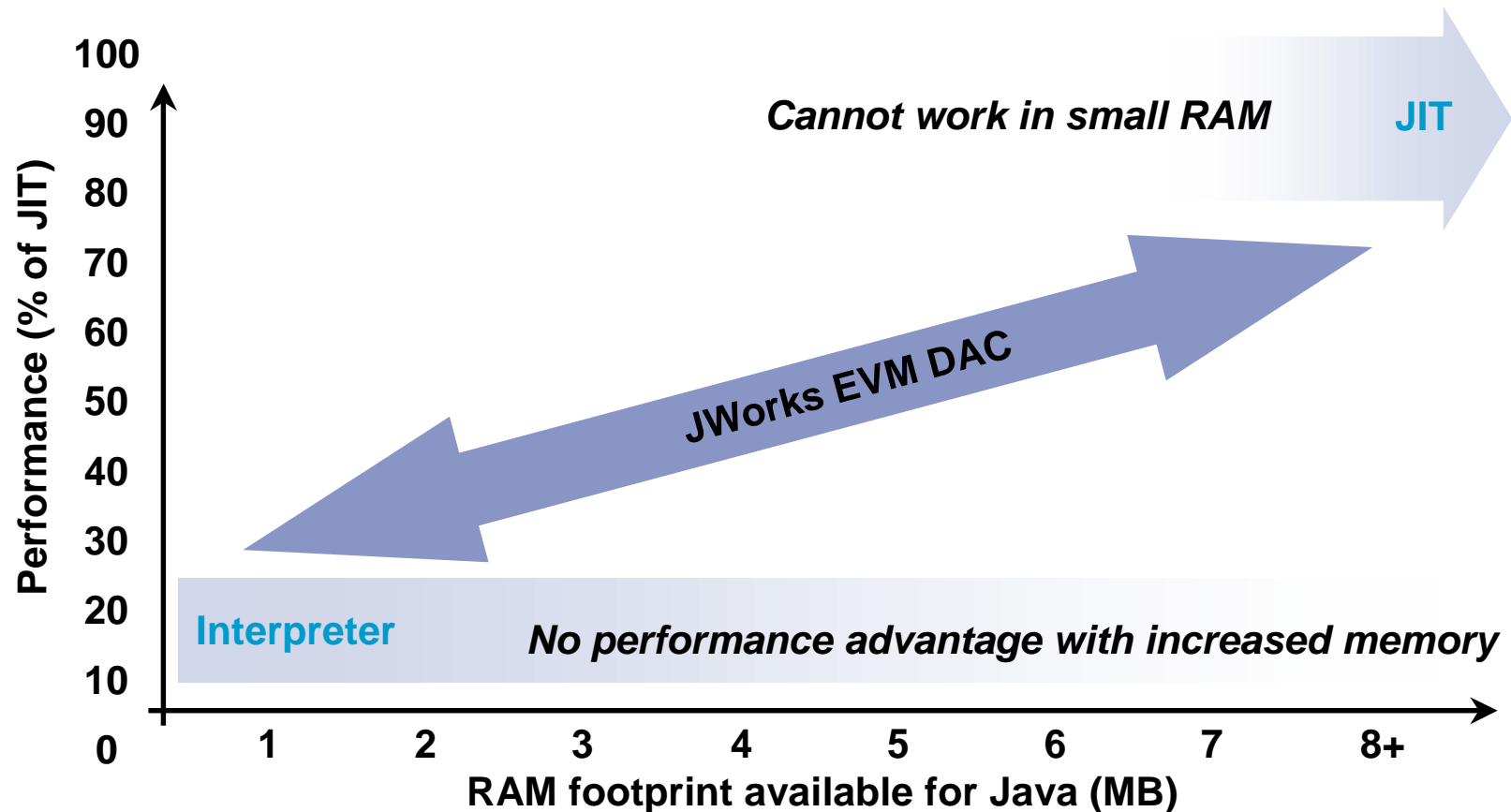
❑ **Solution**

- **Specially-engineered Java solution to meet embedded requirements**
 - Limited system resources
 - Memory size, processor performance, display capabilities, network bandwidth
 - Specific graphical user interfaces
 - Potentially no disk or large storage space
 - Drastic reliability and security requirements
- **Do not spend time on embedded or Java basics!**
- **Integrated, validated components**
 - Invest in product differentiation
 - Get to market faster





Tuning for Performance and Size





Java Performance

- ❑ **Acceptable performance from embedded Java applications is the major impediment to its adoption**
- ❑ **Performance is usually at the cost of size**
 - **Interpreted-only: smallest size but slowest**
 - **Ahead-of-time compilation: fastest but 5x ROM size and loses dynamic extensibility**
 - **Just-in-time compilation: fast but unbounded compilation space**
- ❑ **Solution = dynamic adaptive compilation**
 - **Fast**
 - **Trade-off between speed and size**
 - Compiles only application performance bottlenecks through statistical analysis
 - Memory-bounded
 - Tunable





Improving & Optimizing Performance

❑ VM & CLDC optimizations

- **Compacting Garbage Collector in 1.0.2 CLDC**
 - To improve memory management & reduce heap fragmentation
- **Faster byte code interpreter**

❑ Software acceleration

- **Ahead of Time, Just-In-Time (JIT), Adaptive Just-In-Time**
 - JIT possible for KVM too

❑ Hardware acceleration

- **80% hardware execution - the most commonly used byte codes**
 - Nazomi - JSTAR
 - inSilicon Corp - JVXtreme Accelerator
 - ARM- Jazelle
 - Zucotto Wireless - Xpresso application co-processor
 - Others...

❑ Both hardware and software solutions!



Constrains on Wireless/Embedded Applications

❑ Limited by the Device

- Memory
- Power & Processing Power
- Input
- Screen

❑ Limited by Wireless Environment

- Unreliable, Low-Bandwidth, Expensive
- More Network Errors than in wired lans
- High Risk of disconnection or parameter change because of mobility

❑ Important Questions

- How is the Application impacted by the Limitations
- How important is platform independency
- Security and Management



Java: The Ultimate Wireless Platform?

❑ WAP lessons

- Technical problems: markup language, bearer support, interoperability
- Marketing hype vs. reality

❑ Bluetooth lessons

- (See WAP) When will consumers see a benefit?

❑ Java - a critical complementor

- Java adds function to content to enhance WAP
- But WAP is necessary for Application Provisioning in a GSM environment

❑ Issues

- Fragmentation of Profiles & OEM extensions
- Time to demonstrate application interoperability
- Need to prove it with market demand



Goals of Future Java Technology

- ❑ **Evolving the architecture to enable “everywhere” computing**
 - Improving performance
 - Designing for scalability
 - Enhancing functionality
 - Service Discovery
 - Connectivity
 - Device Management
- ❑ **Driving industry to adopt Java as an end-to-end platform**
 - Residential Gateways
 - Vehicle Information & Entertainment Systems
 - Wireless: 3GPP: MExE / VHE
- ❑ **Building a network effect & growing the value chain**
 - Leverage 1 M Java Developers
 - Serve 1 B Users



Wireless Java Evolution Summary

- ❑ **Content & services drive market requirements**
 - Benefits of Java are clear
 - Value chain is geared up and ready to go!
- ❑ **Segmentation is important - a variety of devices will emerge**
 - Is handset a long-term investment or a consumable?
 - Is performance an issue in light of hardware acceleration?
 - Is the issue application footprint, bandwidth, or both?
 - Promised bandwidth will be slow to materialize
 - Developers will always push for more resources
- ❑ **Competition is increasing - there is no ultimate platform**
 - But Java looks the most promising!



Embedded Java @ WindRiver

❑ Java is strategic for Wind River

- Enabling next generation connected re-programmable devices
- Key technology in many targeted applications

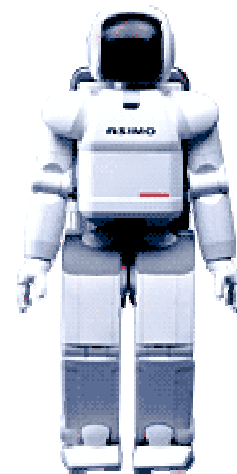
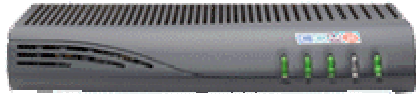
❑ Wind River is committed to Java

- **Industry leadership**
 - #2 JVM vendor after Sun (VDC)
 - Close partnership with Sun & Insignia
- **Community leadership**
 - Participation in JCP and 4 JSR expert groups
- **Technology leadership**
 - Committed to delivering industry-leading solutions to our customers



❑ WindRiver knows Java

❑ WindRiver knows Embedded



How smart things think™

